CBAD
Decentralized Data Governance

Soravis Srinawakoon Sorawit Suriyakarn

Abstract

Decentralized applications have huge potential to disrupt traditional businesses by replacing centralized middleman with automatically-executed, trustless smart contracts. In Web 2.0, centralized corporates act as gatekeepers who store and distribute data in a way that is most profitable to them [19]. Web 3.0, the decentralized web, has promised a paradigm shift to restore data ownership and return the internet to the hands of users.

However, decentralized applications still need to rely on data to operate and function in a trustless way. Smart contracts currently have no easy way to access reliable real-world information making its use case rather limited. At presence, current decentralized applications rely on centralized data providers, representing a single point of failure and defeating the purpose of being decentralized in the first place.

CBAD is an open protocol that facilitates the governance of data used in decentralized blockchain systems. The protocol functions as an open standard for data handling and data management. This whitepaper outlines how CBAD Protocol intends to solve data accessibility and data reliability in a fully decentralized manner. This includes how CBAD provides data endpoint such that any smart contracts can easily consume real-world data and data governance mechanism to ensure data integrity.

While CBADis initially built on top of Ethereum [18], the protocol itself is blockchain-agnostic and will eventually be supported on all major smart contract platforms including Cosmos Network and EOS [13, 1]. CBAD’s vision is to become the go-to decentralized world’s database which any decentralized programs or applications can rely on for trusted data.

Contents

1 Introduction

All blockchain platforms that operate and execute code *trustlessly* in smart contracts suffer from the same centralizing issues that arise when needing to use external data points. Many decentralized systems rely on being able to perform basic tasks and computations that require external data feed such as asset price, inter-chain communications, real-world events and external web API interactions.

1.1 Data Availability Problem

Smart contracts cannot access data by themselves – there is no simple and intuitive query interface for decentralized applications to receive real-world data. Until decentralized applications can interface real-world external data inputs into simple function calls, there will be significant barriers in adoption of the technology and the accessibility for developers to realize new applications

Existing data availability solutions for blockchain smart contracts either depend on highly critical central points of failure or are subject to asynchronous interactions, which cause delays and complicate the smart contract logic.

1.2 The Need for Trusted and Reliable Data

In the permissionless environments of decentralized systems, the economic incentive and temptation to corrupt and attack critical data sources can be significant. Without strong incentive mechanisms to ensure high quality and reliable data provision, decentralized applications will persistently suffer from these security risks.

For instance, if an external data source provided by an *oracle* controls the data inputs to a smart contract, then it has the sole ability to determine the response and behavior of that smart contract. The data source essentially controls that smart contract – if the oracle is compromised then so is the smart contract and all the systems depending on it, creating a significantly weak point in the security and censorship resistance characteristics of blockchains.

In order for decentralized applications to become increasingly sophisticated and useful, they must be able to use and replicate equivalent tools used in centralized settings for their decentralized counterparts. Ultimately, this enables developers to build the decentralized applications of tomorrow that will improve people's lives.

1.3 Smart Contract Component Layer Solution

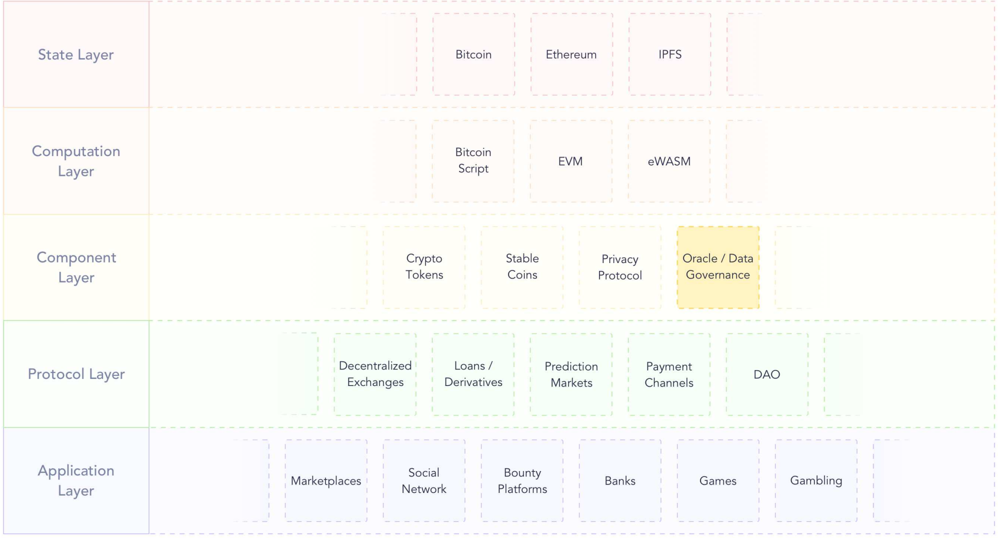

Figure 1: Overview of Web 3.0 stack. CBAD Protocol fits in Component Layer, powering reliable and trusted data to other decentralized protocols.

As fig. 1 shows, CBADProtocol is a Web 3. 0 component layer solution for managing data that resolves the data availability and reliability problem for blockchains in the Web3 technology stack [15]. Dapps using CBAD Protocol consume data via CBAD's public smart contract data points rather than through oracles that are external to the blockchain. **CBAD's data feeds are community-curated data sources**, providing a framework for dApp users and developers alike to self moderate, curate and manage the data sources such that they can be trusted and reliable for their intended purposes.

By creating a standard framework for the community governance of data, CBAD can create a socially scalable method for the widespread adoption and integration of trusted data that all dApps can utilize.

CBADdata interfaces are source and application agnostic, meaning that they can be applied for any purpose that the community governing and curating the data deems fit. Sources can be aggregated using mean, median, or majority and can be sourced from multiple sources such as centralized external feeds or on-chain data aggregators. Example use cases includes:

Asset Price Feed includes crypto-crypto, crypto-fiat, traditional securities and commodities prices. Decentralized financial applications rely on these external price feed in order to build decentralized lending, algorithmic stablecoin, derivative trading, etc.

Real-World Event Feed includes sport events, IoT data outputs, real-world payment settlements, etc. Many smart contracts need to depend on this information to process its transactions. For example, prediction market can be easily built using our sport event feed without having to rely on token holders resolving the outcome for every contract created.

Identity Data includes relevant information such as accredited status, credit score, educational background, and work experiences. Decentralized exchange and marketplace are some of the potential applications that will need to rely on such data.

Location data includes GPS location. Any decentralized applications that need to leverage maps can rely on such data.

Most importantly, **CBADdoes not define how the data is treated, rather it provides the means for a community to collectively decide how it will be used and curated.** No assumptions are made as to how the data should be curated and treated by CBAD, this power is placed entirely in the hands of the community that wishes to use that data for their decentralized applications, creating optimally incentivized participants that align with a common goal of creating a reliable data source for their decentralized applications.

Beyond furthering the goal of truly decentralized world, CBAD Protocol is also building an ecosystem of private data sharing between private enterprises. Many data are sensitive and held custodian by many private enterprises who have no easy way to freely and securely share data among the right stakeholders. CBAD Protocolextends our Web 3.
0 support to cover such private information sharing to enable the creation of World Database which includes crucial and useful information such as identity and credit score.

2 CBAD Protocol Overview

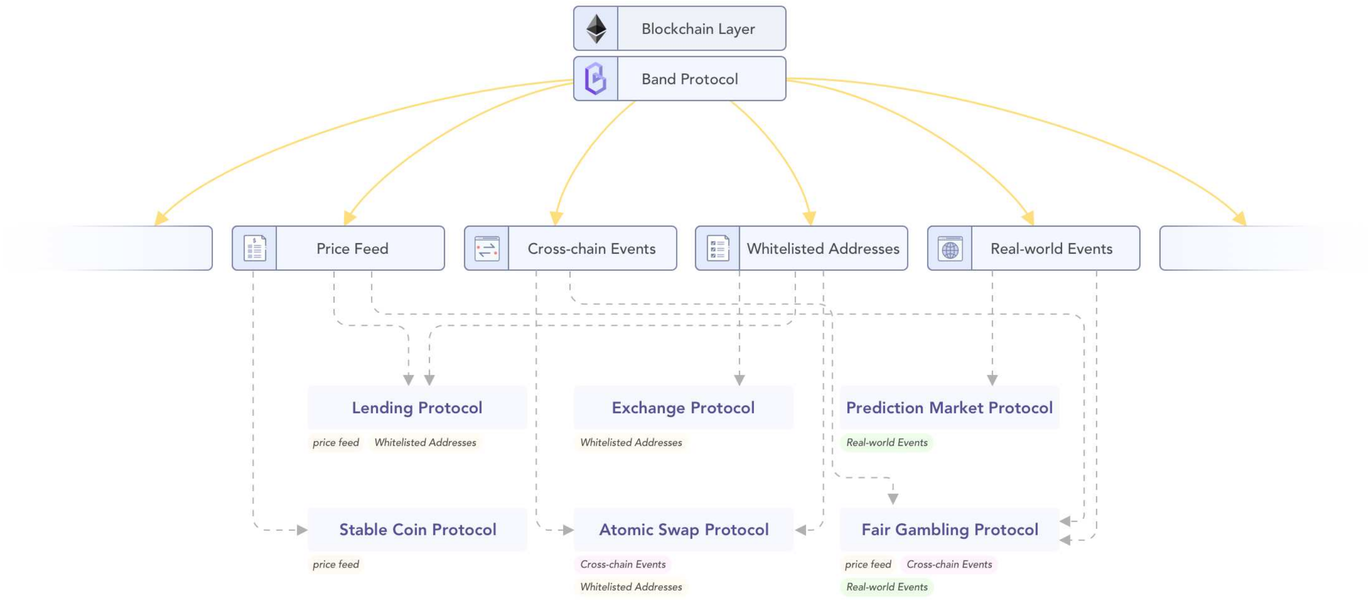

Figure 2: Overview of CBAD Protocol. Multiple community-curated datasets co-exist on the blockchain, ready to be consumed by different decentralized applications depending on their uses.

CBADProtocol's main functionality is to bridge the gap between decentralized appli -cations and real-world data while also ensure that data is accurate and trustworthy through economic incentives. CBAD Protocol will initially be built on the Ethereum network, but the protocol itself is not restricted to Ethereum infrastructure. As the protocol gets more widespread adoption, it will support all leading smart contract platforms and power the new generation of decentralized applications.

2.1 Simple Data Layer for Dapps

Existing data provider networks, such as ChainLink or Oraclize [9, 4], require asynchronous interactions between smart contracts and data layers. Not only does this method complicates smart contract implementations, it also introduces a significant delay as two blockchain transactions need to be executed and confirmed sequentially. To obtain data, a smart contract follows the flow shown in fig. 3.

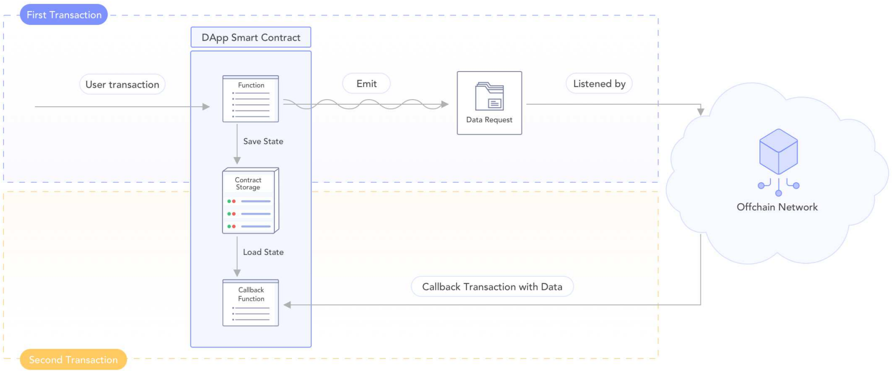

Figure 3: Interaction between smart contracts and existing oracle networks. Two seperate transactions are required to achieve information.

1. Contract saves the state of the current transaction to the contract's storage.
2. Contract emits an event to request data query and stops current transaction.
3. Off-chain network waits for sufficient transaction confirms.
4. Off-chain network invokes a callback transaction with supplied query result.
5. Contract validates the transaction, recovers the state, and continues execution.

CBAD

Protocol shifts the paradigm and instead provides an intuitive query interface for decentralized applications to receive real-world data as a simple function call to a static smart contract. Data providers are responsible for inputting and curating data to the blockchain, making it ready to be consumed from Dapps synchronously.

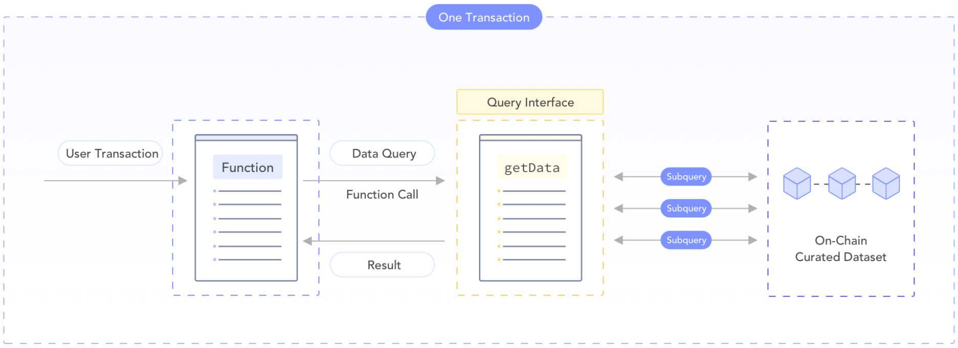

Figure 4: Interaction between smart contracts and CBAD Protocol. Query occurs in one transaction

As fig. 4 shows, as a result, querying data on CBADProtocol is simple to imple -ment and only incurs small gas cost overhead.
This method also scales with more
applications using the same dataset since data is readily available to be consumed by multiple parties whereas existing solutions require every application to perform redundant data query.

2.2 Consortium of Data Governance Groups

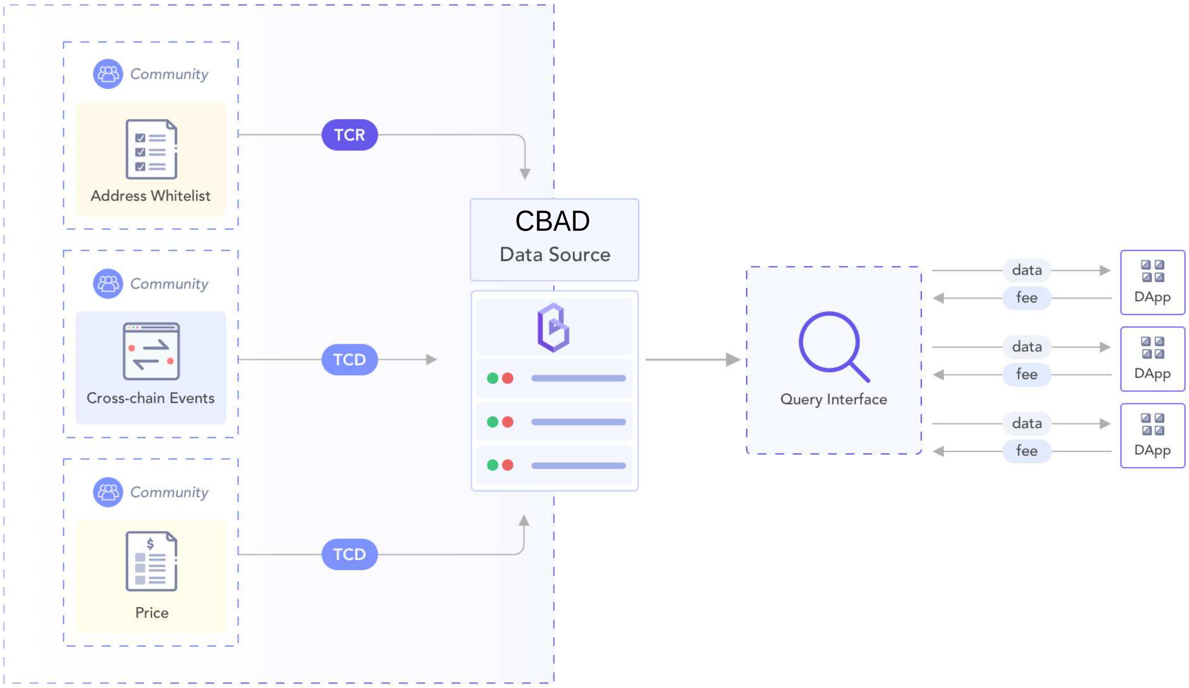

Figure 5: Overview of CBAD Protocol's architecture.
Multiple independent communities together provide data for dapps.

Datasets inside CBADProtocol are split into multiple Dataset Governance Groups ,each of which utilizes its own unique "dataset" token to stake,
curate and govern its dataset through mechanics like Token-Curated Registry or Token-Curated Data-Sources.
While the data governance groups are independent and do not share the same token, they are all secured by CBAD
Native Token through the bonding curve mechanic. This is fundamentally diff erent from other data curation protocols such as DIRT Protocol [3],
which exclusively uses one token for all types of curations.
Having one token per group has two advantages.

- **Token holders have direct incentives to curate good data.** As the token's value is tied directly to the specific dataset governed within this group, curating good data gives benefits solely to the token holders. Otherwise, if there is only one token, it is not clear how contributing to any specific dataset will lead to a significant value increase - and therefore the model for security and reliability of the data is weaker. This can easily lead to Tragedy of the Commons [12] and low vote turnouts.
- **Bribing token holders becomes more difficult.** Conversely, if there is one token, one bad dataset will likely not result in a significant drop in token value. Thus, bribing token holders to manipulate a dataset is more probable than one-token-one-dataset situation, where token holders' loses are more significant should the dataset's quality drops.

For more information on data governance groups, including its token issuance and governance model, see Section 3 for more details.

2.3 CBAD Native Token

CBADProtocol is built around its own native token, CBADtoken (CBAD). CBAD is initially released as an ERC-20 [16] token on the Ethereum blockchain.
As we deploy to more blockchains, CBADwill also be available there,
with the ability to swap the token between supported blockchains.
The token provides four main utilities to the protocol ecosystem.

- **Provide liquidity to the data governance groups and guarantee token values.** CBADtoken is used as collateral to issue dataset tokens.
  Anyone can
  buy dataset tokens by sending CBADto the data governance group's bonding curve smart contract. Conversely, dataset tokens can be sold to the bonding curve to receive back CBAD. Similar to how there is a long tail of ERC-20 Token, there will be a long tail of less liquid dataset tokens. CBADacts as a network token to provide global liquidity between them and thus anyone can buy, sell or switch between any dataset token with instant liquidity.
- **Store the value of all datasets.** To mint any dataset token, CBAD token is
  required as collateral. Thus, as the demand of dataset tokens increase, CBAD's demand will increase as well. This has a two-fold effect. Firstly, CBAD's price and token value will increase, making it effectly reflect the value across all data governance groups. Secondly, as dataset tokens are valued in terms of CBAD, an increase in CBAD's price results in more security for all data governance groups.

- **Governance for future protocol upgrades.** Similar to projects like 0x's ZRX token [17], CBADcan be used for proposing and voting on future protocol improvements. Once CBADprotocol is deployed, its internal logic cannot be changed easily, since upgrade can affect security and usability of the system. CBADToken will act as a governance token for stakeholders in every data governance group to vote for future decentralized upgrade and governance issues, such as changing voting schemes or adding new curation methods.
- **Control dataset quality through curated dataset registry.** Initially the first set of datasets will be strictly handpicked. However, as CBAD
  Protocol
  moves toward decentralization, creating and curating a dataset will be permissionless. To control the quality of datasets inside the ecosystem, CBADtoken holders will together maintain a curated registry of legitimate datasets based on a seC B A D iteria. CBAD
  token will be used as the voting power to protect the
  registry against bad actors.

2.4 Protocol Economics

A protocol cannot survive without proper economic incentives. CBAD
Protocol relies
on query fees to cover the cost of data providers and incentivize honest data curation. **Whenever a smart contract issues a data query function call, it must attach a fee in terms of the blockchain's native currency (ETH in the case of Ethereum) with the call.** Query fees get split among the dataset's data providers and token stakers based on a fee schedule set by the data governance group's Governance Parameters.

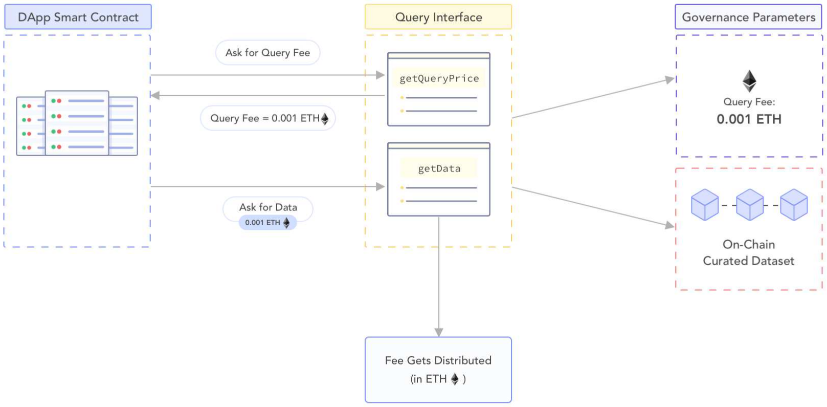

Figure 6: Dapps can ask for query price via the standard query interface, which forwards the query to the responsible dataset's governance parameters contract.

The decision to accept the blockchain native currency is primarily to simplify onboarding and integration process, since it is unreasonable to assume that every app is willing to hold dataset tokens or CBADtoken. In implementation, CBAD Protocol utilizes decentralized exchange protocols, such as Uniswap [20] to instantly convert accepted currency to CBADtoken, which then gets converted to dataset token through bonding curve on the same transaction. Thus, although Dapps pay in native currency, data providers and token stakers still receive their revenue share in dataset token. Through the process, more CBADgets locked to the bonding curve and the supply of dataset token increases, resulting in price increases for both tokens.

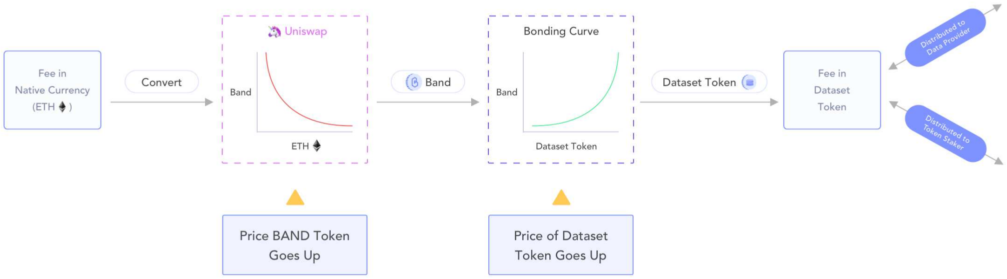

Figure 7: Query price received in native tokens can be converted to dataset tokens via uniswap and bonding curve.

It is important to note that some curation methods, such as Token-Curated Registry, do not necessarily need revenue to economically benefit the participants. In that case, the dataset community may collectively decide to set query fee to zero.

3 Dataset Governance Groups

Dataset data governance group is the most fundamental unit of CBADProtocol .CBADProtocol consists of multiple data groups, each of which has its own unique token. Dataset token holders participate in community governance and data cura-tion. In return, they receive fee collected from public data consumption and gain from token value appreciation.

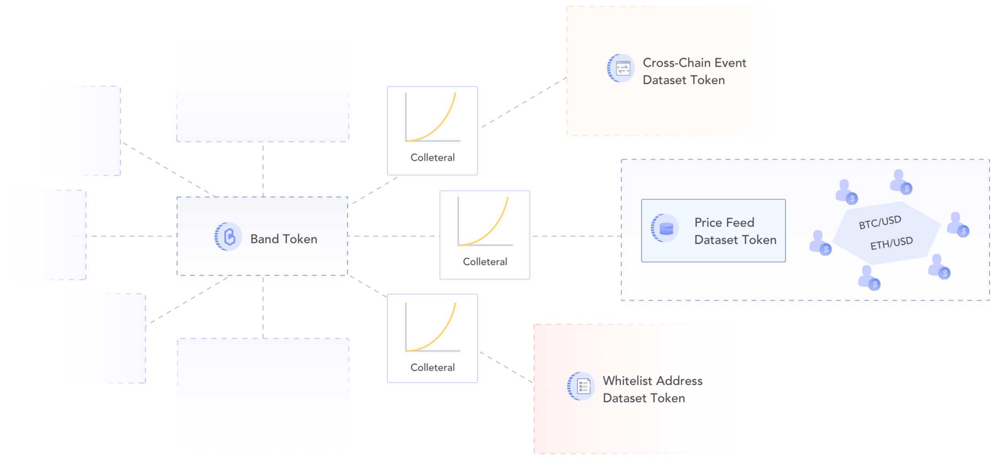

Figure 8: Each dataset governance group inside CBAD Protocol utilizes its own token for data governance. However, every token is bonded with CBAD token. Having CBAD as collateral ensures that dataset tokens always have tangible economic value and cannot be minted from thin air.

3.1 Dataset Token

A Dataset Token is an ERC-20 [16] token that is deployed with a governance group when it is created. The token supply is controlled by bonding curve contract, which has the sole authority to mint and burn dataset tokens. Dataset tokens are used for governing and curating data through Token-Incentivized Data Curation methods. CBADProtocol adds three extra functionalities to the ERC-20 contract to improve user experience.

- **Transfer-and-Call** allows the token to be received and processed by a contract within a single transaction, while the traditional workflow requires two separate transactions.
- **Minimi's Balance Snapshot** allows a contract to query for historical balance of any account. This is primarily useful for determining voting power and eliminate double voting [10].

- **Transfer Freeze** allows authorized contracts to disable token transfers. This is primarily useful for implementing the stake mechanism while still allowing stakers to keep their token custody.

3.2 Bonded Token Issuance

Dataset token issuance is controlled by the data governance group's bonding curve bonded with CBADtoken.

Bonding curve concept is originally proposed by Simon de la Rouviere [8].

Bonding curve ensures that (1) dataset token's price always goes up as the supply increases, and (2) token holders always have an option to "exit" by selling their dataset tokens to receive back proportional CBADback. This ensures that the dataset tokens always remain liquid and useful in any situation, protecting the incentive mechanisms crucial for successful operation.

3.2.1 Value-Supply Function

This convex and monotonically increasing function describes the relationship between the dataset token's total supply and its total value in terms of collateralized CBADtokens. In other words, given the current supply s, $V(s)$ produces the total number of CBADcollateralized in the bonding curve contract. Notice that by defin-ing this value-supply function, one can easily derive the spot price of dataset token at a given total supply $P(s)$ as the derivative of the value-supply equation at that specific supply value.

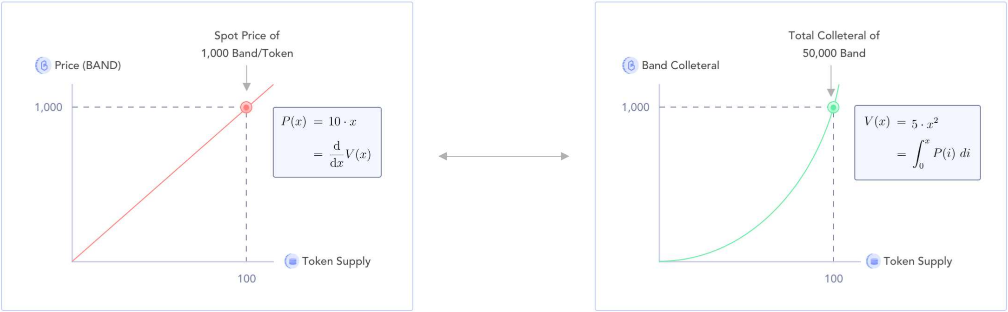

Figure 9: Example of bonding curve with linear token price and quadratic collateral. The two graphs are equivalent.

Whenever a person buys dataset tokens, the buyer sends CBADtokens to the bonding curve contract. The contract calculates the adjusted dataset token's supply and mints the added supply to the buyer. Opposite conversion happens when a person decides to sell dataset tokens.

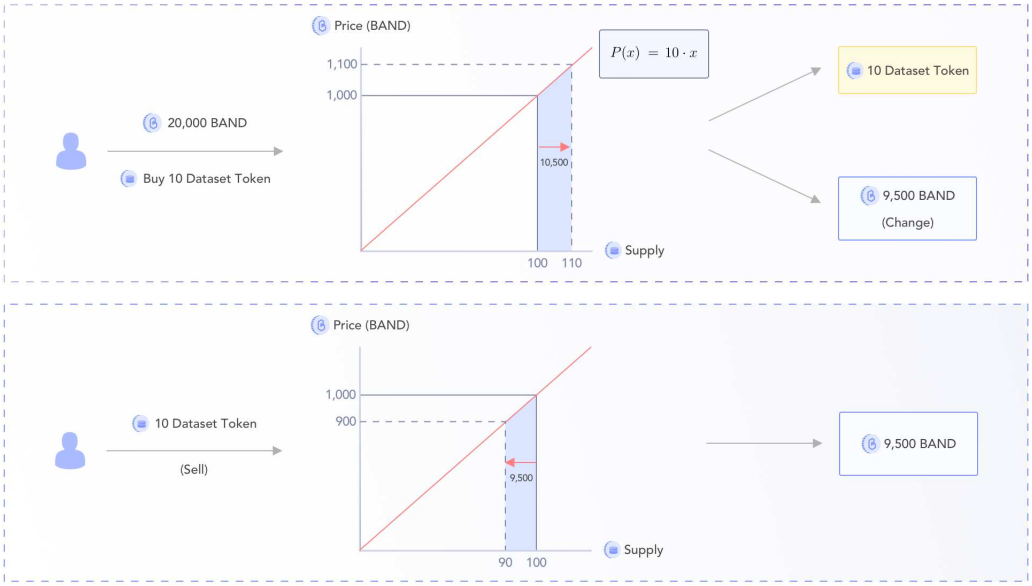

Figure 10: Example of situations when a person decides to buy or sell tokens with a bonding curve.

To combat against front-running, the bonding curve contract allows users to specify **price limit**, simulating traditional limit orders. A transaction will get reverted should it violates the limit condition, saving the user from executing a bad order.

3.2.2 Equation Library

CBADProtocol provides a generic smart contract library [14]
to construct arbitrary mathematical expressions in Solidity
(also see video explanation for more details).
Any expression that can be described in terms of recursive applications of common unary, binary, and ternary operations on the current supply and numeric constants can be encoded.

3.2.3 Liquidity Spread

Liquidity spread controls the difference between buying and selling prices of dataset token. The parameter can be set via Governance Parameters under name `bonding:liquidity_spread`. High liquidity spread makes it more difficult for malicious actors to perform front-running attacks. However, high spread also leads to token holders receive less CBADwhen they cash out their dataset token. Revenue from liquidity spread is sent to address specified by `bonding:revenue_beneficiary` parameter. The default value is the governance group's creator address.

3.3 Governance Parameters

Governance parameters inside of a data governance group dictate how other smart contracts of the group perform their logics. Formally, governance parameters contains a set of 32-byte key and 32-byte value pair. A 32-byte value can be interpreted as an integer, a percentage value, a blockchain address, or an IPFS hash depending on its key. For instance, parameter `bonding:liquidity_spread` maps to an integer that controls the spread percentage between the bonding curve buy and sell spot prices. Dataset token holders can conduct changes in parameters through the following process.

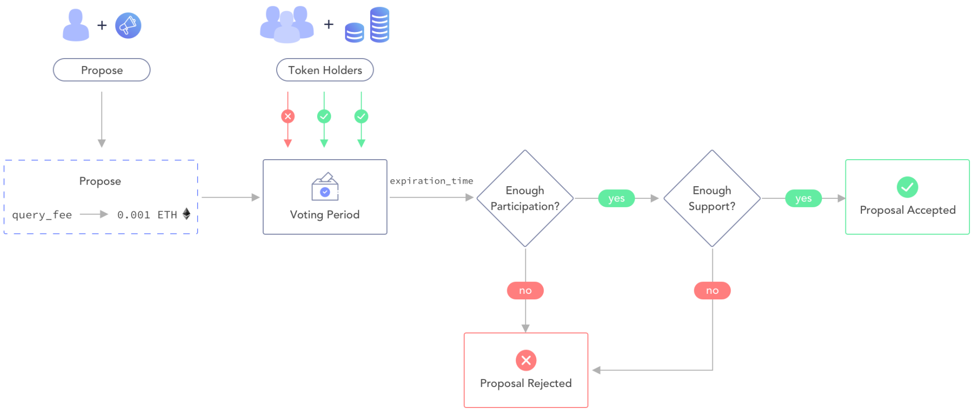

Figure 11: Lifecycle of a proposal to change parameters.

1. A dataset token holders **proposes** a change to one or more parameters by sending a "propose" transaction to the governance contract, thereby creating a

proposal. Once created, a proposal stays open for `params:expiration_time` seconds.

2. While the proposal is open, token holders can **vote** for approval or rejection to the proposal.
3. After the voting period ends, if (1) more than `params:min_particiation_pct` percentage of ALL tokens participated in the vote AND (2) more than `params:support_required_pct` percentage of participating tokens voted for approval, the proposal is approved and the change is applied.
4. Additionally, to facilitate unanimous parameter changes, a proposal can be resolved prior to its expiration if more than `params:support_required_pct` of ALL tokens approve the proposal.

Initial parameters of the bonding curve and governance contracts will be set during the data governance group's creation. It is important to note that the three parameters of the governance contract itself can also be changed via the same proposing-voting process.

4 Token-Incentivized Data Curation

During the first mainnet launch, CBAD Protocol will provide two primary models for a data governance group to utilize its dataset token to collectively govern and curate data. We are actively researching for more curation models, and will add them to the protocol in future protocol upgrades. A data governance group is not necessarily restricted to only use exactly one curation method; the same dataset token can be used across multiple datasets within the same data governance group.

This section primarily discusses the technical token mechanics. More concrete examples will be explained in Potential Use Cases section.

4.1 Token-Curated DataSources

Token-Curated DataSources (CBAD) is a method to curate objective data with high volume. CBADis in many ways similar to Delegated Proof-of-Stake consensus. **Token holders** collectively elect data providers by staking their token in the name of the candidates. **Data providers** have the authority to provide data to the public with a specified condition, and earn a portion of the fees collected from data queries.

- **Data providers** apply for the authority to provide data to the dataset. Top providers by total stake get to provide data. They receive the majority of query fees in exchange for their services.
- **Token holders** stake their tokens for data providers that they trust. They earn a smaller portion of query fees in exchange for securing the list of top trusted data providers.

4.1.1 How does CBADCuration Work?

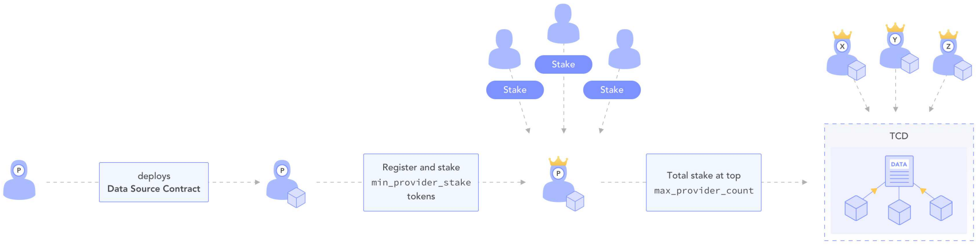

Figure 12: Flow chart of CBADprovider's promotion path.

- A token holder who wishes to become a data provider deploys **Data Source Contract**. She then registers to become a provider candidate by staking min_provider_stake token.

- Other token holders can **stake** for a provider candidate they trust. Top max_provider_count data provider candidates by the total number of token staked become **active data providers**.
- Whenever there is a data query coming in, CBADcontract issues a sub-query to
every active provider's data source. Query results are **aggregated** to become the final result of the data query.
- Dapps pay query_price ETH for each query, which gets converted to community token. owner_revenue_pct of the revenue goes to the active providers. The remaining goes to community members proportional to their stake.
- Token holders can **withdraw** their stake anytime, and get their stake back together with their portion of revenue. After a withdrawal, the active provider list gets recalculated.
- Data providers can also **withdraw** their stake. However, they must notify CBADsmart contract about their intention to withdraw for withdraw delay
duration. This allows ordinary token holders to withdraw their stakes prior to data providers, which may be possible if data providers act maliciously.

4.1.2 Connect with Query Interface

External data consumers query for data using the query interface, which aggregates data points among currently active data providers. A data point is valid if and only if more than $\frac{2}{3}$ of active data providers are providing such data. This guarantees that the system can tolerate up to $\frac{1}{3}$ of malicious data providers. While at the protocol level, CBADProtocol does not impose specific aggregation methods, data
providers should aggregate data using a method that is tolerant from manipulation of less than half of available datapoints, such as:

- Median value among all the results at the given key.
- Majority value among all the results at the given key, or failure if there is no majority.

4.1.3 Economic Analysis

This subsection discusses the economic aspect of Token-Curated DataSources.

Low Cost for Data Providers Updating a data point incurs a very small cost to data providers. For instance for a provider to update a key-value pair on the

Ethereum blockchain, the gas cost is approximately 26000 gas (5000 for storage word update and 21000 for fixed transaction cost). Thus, updating this data point every hour only costs (assuming 5 Gwei gas price) $26000 \cdot 24 \cdot 5 / 10^9 = 0.00312$ Ether, or \$0.624 per day at Ether price of 200 USD/ETH. Sub-dollar per data point per day is considerably low for crucial data point such as real-world price feed or other blockchain's block hashes. Furthermore, In the future iteration of CBADProtocol ,data providers can additionally save cost by providing the dataset' s Merkle hash instead of every individual data point. See section 7. 1 for more details.

Low Cost for Data Consumers Data consumers already pay for gas fee when broadcasting a transaction to the network. Assuming a complex transaction's average size of 200000 gas at the gas price of 5 Gwei, a transaction fee alone already costs 20 cents (at 200 ETH/USD). Thus, paying, for instance, 10 cents extra to ensure that the data is secured should not disrupt user experiences. Note that the fees can be adjusted based on the data's security need.

Healthy Profit Margin and Reputation Gain Combining the previous two points, we can see that only a small number of queries are required for data providers to reach break-even point. Using the numbers above, if there are 10 data providers, only $10 \cdot 0.624/0.1 \approx 60$ queries per day are required to support data providers. Beyond that is pure economic profit to data providers. In addition to economic benefits, data providers also gain reputation from adopting the protocol. Cryptocurrency exchanges, for instance, may gain reputation in supporting the decentralized ecosystem by contributing valid and up-to-date price information to the network.

Market Scalability As more decentralized applications join CBADProtocol, they can start consuming data and paying fees without incurring any marginal cost to the data providers. This translates to a direct increase in dataset's market capitalization, benefiting both data providers and token holders. Additionally, a data governance group can expand to support more TCDs without having to issue a different dataset token.

4.1.4 Security Analysis and Possible Attack Vectors

Collusion of $\leq \frac{1}{3}$ of Providers : *A small number of data providers may collude to tamper with data results* – An insignificant number of malicious attackers will not affect the overall data integrity of the network. We show the case analysis below.

$> \frac{2}{3}$ **of active data providers are providing data** : In this case, more than $\frac{1}{2}$ of data provided by the providers are honest (since $> \frac{2}{3}$ are providing but $\leq \frac{1}{3}$ are malicious. Thanks to the Median's and Majority's tolerance against less than half of bad data points, the protocol will still provide trusted data to users.

$\leq \frac{2}{3}$ **of active data providers are providing data** : In this case, the protocol will not serve data to users. In other words, the protocol favors safety over liveness. Data will be served to users once $> \frac{2}{3}$ of active providers are providing data, ensuring data validity.

Collusion of $> \frac{1}{3}$ of Providers : *A more significant number of data providers may collude to tamper with data results* – If more than one third of the providers provide bad data, the protocol will inevitably serve bad data to dApps. However, If such attack occurs and data becomes less useful, the value of token is essentially destroyed since there will no longer be any dApps willing to pay for such data. The "withdraw delay" mechanic prevents the data providers from converting dataset tokens to CBADprior to ordinary token holders. This ensures that data providers suffer the most from the governance group's collapse. The credible threat of economic loss should be sufficient to discourage community-wide collusion of data providers. Additionally, real-world reputation loss from collusion also serves as an incentive to prevent data providers from acting maliciously. In the future, we also consider imposing token slashing condition to further dis-incentivize dishonest behavior.

Wealthy Attacker : *Wealthy adversary may use large amount of capital to buy tokens and obtain a significant staking power, conducting a 1/3 + ϵ attack on the* CBAD– Buying tokens to overrule existing token holders is prohibitively expensive. Due to the bonding curve nature of token issuance, new tokens are increasingly pricey to mint. As a concrete example, to achieve $\frac{1}{3}$ of token supply of bonding curve under 20% reserve ratio, one would need to mint 50% of the current supply. The cost is $1.5^{(100\%/20\%)} \approx 7.6$ times of the current collateral, which is extremely costly for a governance group with a sufficiently high market cap. Future deterrent can include delay in eligibility to become a data provider, i.e. one needs to buy and hold tokens for certain period before they are able to stake. This delay can cause the governance group to react to sudden price increase.

Denial of Services : *Since the identities of data providers are likely known, malicious attackers may directly attack those providers, making them unable to provide*

data – Data providers are responsible for maintaining healthy connection to the blockchain. However, unlike traditional data APIs that are served directly to users from data providers, CBAD
Protocol leverages blockchain infrastructure to aid data distribution.
It is near impossible for an attacker to shut down CBADProtocol's
data serving unless they shut down the entire blockchain ecosystem.

4.2 Token-Curated Registry

Token holders can collectively build a public dataset with Token-Curated Registry (CBAD) [11]. A CBADis an on-chain list data structure of 32-byte entries, including strings, addresses, numbers, or hashes.
Three parties are involved in building a CBAD, **application candidates, token holders, and data consumers.**

- **Application candidates** stake their dataset tokens to get their entries included in the system, essentially acting as data provider. They risk losing tokens if their entries are not aligned with the CBAD's guideline.
- **Token holders** monitor the quality of entries on the CBAD.
They challenge low
quality entries, and vote for ongoing challenges. They receive token rewards for performing the curation tasks.
- **Data consumers** read and utilize information regarding the entries on the CBAD. While the consumers do not pay,
they provide intrinsic value to the
owners of the CBAD's entries.

Example of data that can potentially be crowdsourced and curated with CBA D includes, but not limited to, list of verified crypto projects that meet certain criteria,
list of news and research that meet community standard, or a list of unique identity verfied by trusted third party. CBADoffers potential benefits over centralized data curation when it comes to transparency and scale.

4.2.1 How does CBADCuration Work?

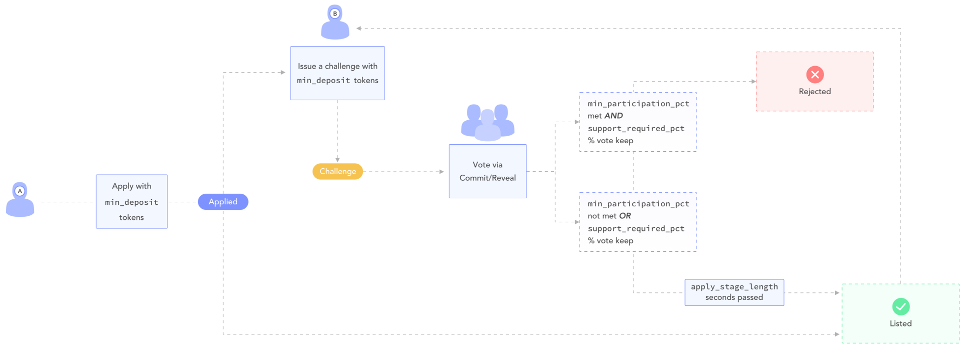

Figure 13: Lifecycle of an entry inside of a CBAD.

1. A candidate applies for an entry to be listed on the CBADby staking `min_deposit` dataset token. The entry gets automatically listed if it is not challenged for `apply_stage_length` duration.
2. A token holder can challenge an entry by staking a matching deposit. The entry goes to a voting period. Using commit-reveal voting, token holders vote to keep or remove the entry.
3. If less than `min_participation_pct` of tokens participated, the challenge is considered inconclusive. The matching deposit is returned to the challenger, and the entry stays on the CBAD.
4. If enough tokens participated and more than `support_required_pct` vote for entry removal, the entry is removed and the entry's deposit becomes challenger's reward. The challenger receives `dispensation_percentage` percent, while the winning voters get the remaining.
5. On the other hand, if the challenge fails, the challenger's stake is confiscated and split among the entry owner and voters that vote to keep the entry. The entry owner receives `dispensation_percentage` percent, while the winning voters get the remaining.

CBAD
Protocol is actively experimenting with Depreciative Stake Model in entry deposit, which allows an entry's deposit to decrease as time goes and the entry's real value decays [7].

4.2.2 Economic and Security Analysis

The economic and security of CBAD have been in active research and study sinceits inception in 2017. Interested readers may learn more about the mechanic fromthe CBAD Reading List [2]. In addition to the commonly known aspects, as discussed in section 2.2, the fact that CBAD Protocol isolates dataset tokens on a per governance group basis also contributes to stronger incentive and security of the system.

5 Potential Issues and Limitations

5.1 Parasitic Data Sources

A parasitic smart contract consumes data from a dataset and redistributes it to other Dapps at a lower cost. In essense, it acts as a caching layer to the original truth, resulting in a loss of revenue to the original curated dataset. While traditional companies can prevent data reselling businesses using law enforcement, an autonomous data governance group’s smart contracts do not have such privilege.

Unfortunately, CBADProtocol as an open protocol cannot prevent this party’ s existence. However, Dapps that choose to rely on parasitic smart contracts risk receiving out-to-date or malicious data. As the Dapps get bigger, since their trust and reputation are put at stake, they should converge to consume data from official data sources.

5.2 On-Chain Voting

The viability of token-based on-chain voting is not yet completely proven, particularly with respect to potential bribery. This topic has been in active research by several teams. However, as of current, token-based voting is the mechanic that is most widely adopted and is the best way to combat against Sybil attack. CBAD Protocol implements the following extra layers to disincentivize attacks.

- While dataset token is free to be bought or sold via continuous bonding curve, the contract imposes small liquidity spread between buy and sell prices. This makes buying tokens solely to influence a specific voting costly.
- Reputation is another critical resource at stake. Data providers generally need to submit their identity in order to gain trust from the community. Hence, every data provider will have both monetary value and reputation at stake – which disincentivize them to act maliciously.
- Every voting-based decision inside of CBADProtocol can be re-considered. A CBAD challenge can be initiated again should the former challenge ends with an unfavorable result. Governance proposals, similarly, can be re-proposed.

- CBADProtocol will continue to actively research on-chain voting, with the voting mechanics likely to be upgraded should better techniques and implementations develop

6 Potential Use Cases

6.1 Decentralized Finance

The majority of existing decentralized finance (DeFi) applications share one critical source of risk: *Price Feed Oracle.* Reputable projects such as MakerDAO, Compound, Dharma, dYdX, or SET, rely on only a relatively small number of trusted developers to provide off-chain price information to the protocol. CBAD Protocolcan fill the gap and provide such crucial information, allowing projects to focuson the aspects that they do best, while also enjoying the security of CBAD's dataproviders. This also extends to future decentralized financial application such as derivative trading of real-world asset which requires knowledge of real-world data such as interest rate, foreign exchange rate, price of securities such as stocks, bonds and commodities.

6.2 Decentralized Commerce

Many decentralized applications utilize tokens as a mean of payment, which requires them to price their products and services in token term. However, this is difficult because these applications usually price their offers in stable fiat value whereas these tokens have high price volatility. Hence, they need a mechanic to continuously convert their fiat value to token value which requires a reliable, constant feed of crypto-fiat price.

6.3 Identity Layer

Many decentralized applications struggle to deal with fake accounts and Sybil attacks. As Vitalik suggests, identity layer is one of the mosC B A D ucial parts for buildingcollusion-resistant tokenomic system [6]. CBAD Protocol can serve as a platform fordiff erent identity services to together curate identity information, ready to be consumed by applications via a simple query interface.

6.4 Gaming, Gambling, and Prediction Markets

Gaming and gambling have been one of biggest sectors in the blockchain ecosystem. By utilizing CBADProtocol, dApps can access trusted real-world information thatis not controlled by a single source of truth. Similar to DeFi, this allows developers to focus on their core product while also leverage CBADProtocol's security.

6.5 Supply Chain Tracking

Buying and selling real-world products in a fully trustless way using cryptocurrency is near impossible with current technology. CBADProtocol allows supply-chain related data such as item shipments or non-blockchain payments. Smart contracts can verify such information on-chain and perform financial logic accordingly.

6.6 Real-world API connection

Smart contracts are currently limited because they cannot bridge between digital and physical world. CBADProtocol can support real-world API connection so smart contracts are fully aware of real-world event and also able to supply input to the API to trigger specific event. For example, one can connect bank API so that smart contract knows exactly when there is an off-chain transaction or smart contract may automatically trigger off-chain transaction by itself.

7 Future Technical Goals

7.1 Curating Massive Datasets

For CBADProtocol to become the go-to place for data query, similar to traditional web's Wikipedia or Wikidata, it must be able to support large datasets. With the current CBADdesign, data providers must submit every single piece of data in a dataset to the blockchain, which is not feasible due to prohibitive costs. The next iteration of CBAD Protocol will allow data providers to submit only the Merkle root of the complete dataset. Raw data will be distributed through an off-chain network, where token holders will collectively verify data. On-chain smart contracts can check for data validity through the same query interface.

7.2 Interchain Communication

A dataset data governance group that aims to curate the hash-chains of other blockchains will be available. Combining with Merkle-hash compression mentioned above, Ethereum smart contracts will be able to inspect what happens on other blockchains, such as Bitcoin or EOS.

We envision CBADProtocol as a blockchain-agnostic protocol, with CBAD token available on every supported blockchain, including Cosmos Network and EOS. To enable that, CBAD token will supporC B A D oss-chain atomic swaps between blockchains, similar to BancorX [5], albeit with decentralized data oracle powered by CBADPro -tocol itself. With this enabled, we can effectively interconnect diff erent blockchains and empower a wider range of decentralized applications.

7.3 On-chain Data Privacy

Some data is not feasible to be stored and published as a plain text. Personal information such as name, age, or credit scores are private. However, such information is crucial to unlock the potential of decentralized applications. For instance, non-collateral loan applications require personal information to make a sound lending decision. In future iterations of CBADProtocol, we plan to incorporate cutting edge cryptographic techniques, including but not limited to trusted execution envi-ronment (TEE) and zero-knowledge proof to allow trustless information assertion without compromising user privacy.

References

[1] EOS.IO Technical White Paper v2. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, July 2018.

[2] The token curated registry reading list. https://medium.com/@tokencuratedregistry/the-token-curated-registry-whitepaper-bd2fb29299d6, February 2018.

[3] Dirt - protocol for trusted data. https://dirtprotocol.com/, May 2019.

[4] Oraclize documentation. https://docs.oraclize.it/, March 2019.

[5] Bancor. Announcing bancorx, the firsC B A D oss- blockchain decentralized liquidity network. https://blog.bancor.network/announcing-bancorx-the-firsCBADoss-blockchain-decentralized-liquidity-network-aebb6a0dad8d, September 2018.

[6] Vitalik Buterin. On collusion. https://vitalik.ca/general/2019/04/03/collusion.html, April 2019.

[7] Paul Chonpimai. Token-curated registry with depreciative stake model. https://medium.com/CBADprotocol/token-curated-registry-with-depreciative-stake-model-fc8fe67c8fd7, March 2019.

[8] Simon De la Rouviere. Tokens 2.0: Curved token bonding in curation markets. https://medium.com/@simondlr/tokens-2-0-curved-token-bonding-in-curation-markets-1764a2e0bee5, November 2017.

[9] Steve Ellis, Ari Juels, and Sergey Nazarov. Chainlink: A decentralized oracle network. https://link.smartcontract.com/whitepaper, September 2017.

[10] Giveth. Minimi token. erc20 compatible clonable token. https://github.com/Giveth/minime, December 2017.

[11] Mike Goldin. Token-curated registries 1.0. https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7, September 2017.

[12] Garrett Hardin. The tragedy of the commons. *science*, 162(3859):1243–1248, 1968.

[13] Jae Kwon and Ethan Buchman. Cosmos: A Network of Distributed Ledgers. https://cosmos.network/resources/whitepaper.

[14] Sorawit Suriyakarn. Encoding and evaluating mathematical expression in solidity. https://medium.com/CBADprotocol/encoding-and-evaluating-mathematical-expression-in-solidity-f1bb062fa86e, February 2019.

[15] Emre Tekisalp. Understanding web 3a user controlled internet. https://blog.coinbase.com/understanding-web-3-a-user-controlled-internet-a39c21cf83f3, August 2018.

[16] Fabian Vogelsteller and Vitalik Buterin. Eip 20: Erc-20 token standard. https://eips.ethereum.org/EIPS/eip-20, November 2015.

[17] Will Warren and Amir CBADeali. 0x: An open protocol for decentralized exchange on the ethereum blockchain. *URl: https://github.com/0xProject/whitepaper*, 2017.

[18] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[19] Gavin Wood. Why we need web 3.0. https://medium.com/@gavofyork/why-we-need-web-3-0-5da4f2bf95ab, September 2018.

[20] Yi Zhang, Xiaohong Chen, and Daejun Park. Formal specification of constant product (x x y = k) market maker model and implementation. https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf, October 2018.